

Deconvolution in Communication Systems

The title contains a fancy name for a DSP process that can reveal details of wave propagation paths without prior knowledge of their nature. Multipath and other distortions may be detected and corrected to some extent with this technique. It's also useful in DSP filter design. Come discover how it's done.

By Doug Smith, KF6DX

Multipath distortion is the enemy of many radio communicators, whether they are interested in moon-bounce, terrestrial microwave or HF. Multipath may be generally described as a situation in which radio signals take many different routes between transmitter and receiver. Those routes quite often have different lengths, so received information consists of a multitude of superposed copies of the transmitted information, smeared over time.

It might seem at first that there is nothing DSP or any other technology can do about that distortion, since it is caused by physical phenomena beyond

our control. But where knowledge exists beforehand about the nature of the transmitted signal, it turns out something often can be done. I'll try to explain what I've learned about that.

Modeling Multipath Environments

Imagine you're standing just inside the Taj Mahal (Fig 1). The clack of shoes meeting tile and the hush of whispers bounce lightly from the walls and ceiling. Your friends entered just moments ago. You scan the foyer; they aren't in sight. You call out to them. Your voice echoes through the halls and chambers of that place for what seems like an eternity—until security personnel come and tell you not to do that!

Your friends, having reached a distant part of the building by now, hear

the sound; but it doesn't sound much like you. In fact, it sounds more like a dull roar because your voice has taken so many paths to their location. All the echoes overlap so much that words and even syllables are indistinguishable. You are in a *reverberant environment*.

Your friends begin walking toward you. As they come closer, you speak again. This time, they understand you and reply. The number of paths and the differences in their lengths have now decreased; the time smearing and overlap of echoes are now little enough to allow you to be intelligible. You have demonstrated a useful model for reverberant environments: many discrete paths, each with its own transit time or delay and each with a particular attenuation. See Fig 2.

In the figure, multipliers h_n have

values less than unity and represent the attenuation on paths whose delays are proportional to n . Notice that no signal propagates directly from the input to the output; the output is derived only from delayed signals. That indicates the usual situation: Your friends are some finite distance from you, even when in view. So even on a direct path, there is always a positive, non-zero propagation delay. The model also applies when your friends are around the corner; they cannot hear your voice directly, but only the sound that is bouncing off the walls, floor and ceiling.

When the delays z^{-n} in the model are spaced apart by the same amount of time, which we shall call the *sampling period*, the set of attenuation constants h_n is referred to as the *impulse response* of the system. In fact, Fig 2 is exactly the same as the block diagram of a *finite-impulse-response (FIR) filter*, a common construct in DSP.¹ While it is perhaps strange to think of the Taj Mahal as a filter, that is indeed what it is. When the constants h_n are chosen strategically, the system may be made into almost any filter shape imaginable. When they are undefined, as in the case of sounds propagating through buildings or radio signals through whatever medium, the transfer function (frequency response) is also undefined.

If the impulse response of a system can be found (the model), then another system may be built having a transfer function that is the inverse (the inverse model) of the original system. When the two systems are cascaded, the final output is a restored version of the input signal (the desired). For the Taj Mahal or a set of radio propagation paths, the hard part is discovering the original impulse response. When the environment is known and fixed (as in the Taj Mahal), the impulse response may be discovered by modeling the structure and doing ray-tracing experiments on a computer, for example. When the environment is unknown (a radio path), we must resort to *inverse modeling* to get a clue about the corrupting system's impulse response.

That is fairly easy when the unknown environment is fixed. When it is changing, it is much more difficult. Even then, though, DSP provides weapons to combat the enemy. Follow me into a discussion of how those two cases are generally handled.

Inverse Modeling

When performing the operation

¹Notes appear on page 51.

shown in Fig 2, the output is the sum of all the delayed, attenuated signals. That sum is called a *convolution sum*; the input signal is said to be *convolved* with the impulse response. A convenient notation for the convolution sum is:

$$r_t = \sum_{n=0}^{L-1} h_n x_{t-n} \quad (\text{Eq 1})$$

where r_t is the output at discrete time t , x_{t-n} is the original input signal at time $t-n$, and L is the length of the finite-impulse response. The transfer function of that system may generally be found by taking the discrete Fourier transform (DFT) of the impulse response, h_n :

$$H_\omega = \sum_{n=0}^{L-1} h_n e^{-j\omega n} \quad (\text{Eq 2})$$

Where ω is the angular frequency in radians/s. The goal of inverse modeling is to discover the system that has a transfer function equal to the reciprocal of H_ω . Were a copy of the original input signal, x_t , available, that would be easy to do, as shown in Fig 3. The corrupted signal r_t forms the input to the inverse filter, whose coefficients are adjusted in some way based on a comparison between the original input signal, x_t and the doubly processed output of the inverse filter, y_t . When the error signal e_t goes to zero, the inverse filter's frequency response G_ω is the reciprocal of the corrupting system's:

$$G_\omega = (H_\omega)^{-1} \quad (\text{Eq 3})$$

Inverse filter G is said to "*deconvolve*" the original input signal and the

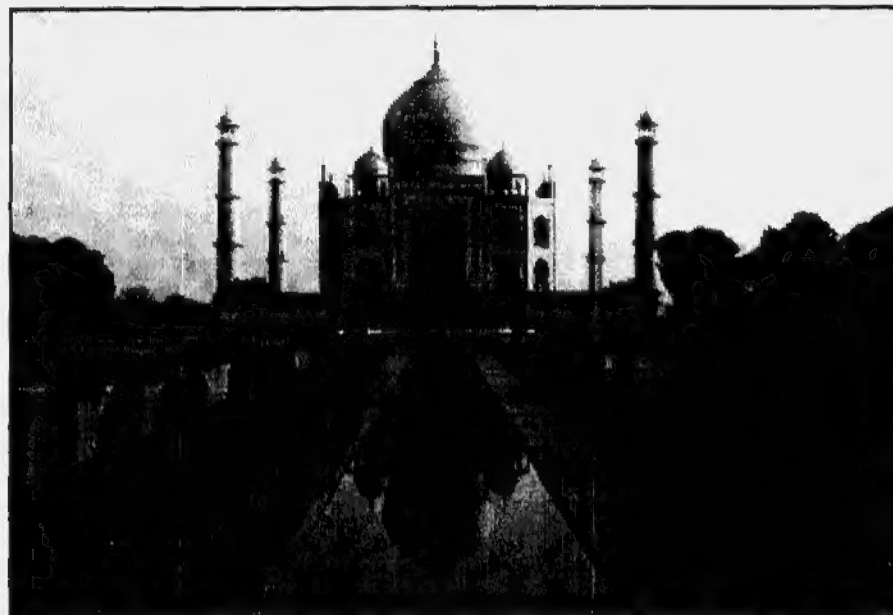


Fig 1—A reverberant environment.

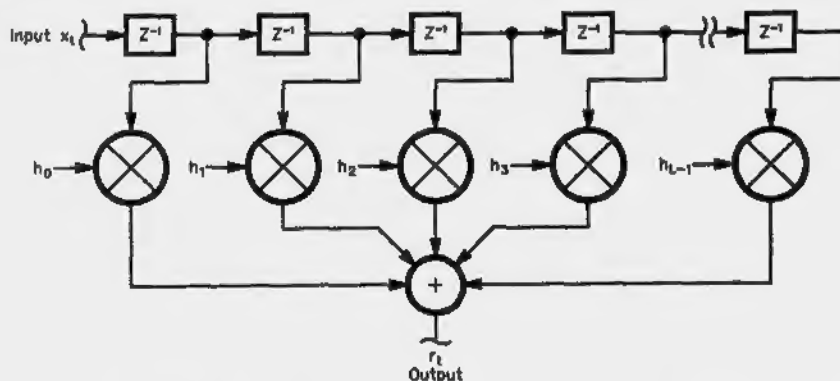


Fig 2—An FIR model of a reverberant environment.

corrupting filter's response, restoring the original signal. In the steady state, the inverse filter performs the operation:

$$y_t = \sum_{n=0}^{L-1} g_n r_{t-n} = x_t \quad (\text{Eq 4})$$

The transfer function of the cascaded system is a single, unity-amplitude impulse. A fixed delay is used in the upper path of the original input signal to compensate the delay through the two filters.

It's perhaps surprising that impulse response g_n is not necessarily the inverse DFT of G_ω , although that has sometimes been stated, incorrectly. The proof of that is a bit more complex than what I want for this article. Oppenheim and Schaffer take up the subject briefly.²

Now it's time to mention how impulse response g_n is adjusted during inverse modeling to efficiently achieve the desired result. The most popular method is called the least-mean-squares or LMS algorithm. It was published by Widrow and Hoff in 1960³ and it's the same as algorithms currently used for adaptive noise reduction and automatic notches in radio receivers.

In the LMS algorithm, each value or coefficient of the impulse response is adjusted at each sample time according to:

$$g_n' = g_n + 2\mu e_t x_t \quad (\text{Eq 5})$$

for L values of n , where μ is a constant chosen to alter the speed of convergence and the amount of error in the steady-state solution. Additional details of the behavior of adaptive filtering systems may be found in the Amateur Radio literature⁴ and will not be treated further here.

You may be questioning how the methods described above can be useful, since they require a copy of the signal originally sent. One application is found in the suppression of echoes on telephone circuits. Another is found in DSP filter design.

Telephone-Line Echo Suppression

On a two-wire, full-duplex telephone circuit, hybrids are used at each end to segregate transmitted and received signals. The hybrids must achieve significant isolation between the two signals lest a signal transmitted at one end arrive at the other end to be retransmitted toward the sender. The result is a series of echoes. When termination impedances are not perfect on the line or imbalance exists, those echoes are always present. They

are most discomfiting to the talker—and perhaps also to the listener—especially over lengthy, overseas paths having transit times of 300 ms or more. This sort of thing can also be a problem in speakerphones.

The system of Fig 3 may be employed to eliminate the echoes, since copies of both transmitted and received signals are present at the transmitter. That is, in fact, what telephone companies currently do to handle the situation. I notice some long-distance companies need to check the operation of their echo cancelers. Echoes were rampant in the early days of long distance, then the problem seemed to have virtually disappeared for a long time; but now, I regularly get reports of its reappearing.

LMS Filter Design

This example is one of direct modeling rather than inverse modeling, so it's a little different from what we've covered so far; but it's still useful because it shows something about the underlying concepts of system modeling in general.

Imagine we want to find the finite impulse response corresponding to some particular filter shape—say, a low-pass. First, we must characterize the desired transfer function, H_ω , completely by both its amplitude and phase responses. Amplitude versus frequency is more important at this stage than phase; we dream up a *pseudo-filter* having the desired response. FIR filters generally have linear phase responses; the phase versus frequency plot is a straight line. We place this pseudo-filter into the modeling system shown in Fig 4. Notice that the pseudo-filter need not actually exist as an FIR filter; it is just a block that replicates the transfer function we want, and we may perform that function in any way.

To make the adaptive filter, G , in Fig 4 converge to match the pseudo-filter's response, the generator signal

x_t 's spectrum must contain energy at all frequencies of interest. White noise is a good first choice for this signal source. Start the thing going and when the LMS algorithm has minimized the error e_t , the adaptive filter will have converged on the impulse response most closely matching our desired response.

Depending on the length (L) of the adaptive filter, it may be difficult to achieve the desired response at certain frequencies exactly. That may be addressed in LMS filter design by changing the amplitude-versus-frequency content of the generator, x_t . A large relative amplitude of the generator's content at some particular frequency allows the filter to more closely meet its specification at that frequency.

When a Desired Response Is Not Available

For radio signals, the telephone scenario above is not particularly relevant. The question becomes "How can we use inverse modeling when a copy of the original signal is not available?" The answer is that the signal x_t used to compute error signal e_t and used in the LMS algorithm need not be an exact copy of the original; it need only be a reasonable approximation of that signal. Any information about the original is useful in nudging the algorithm toward convergence at the start of adaptation; we then get a better deconvolution that, in turn, helps the next iteration toward the optimal solution. Let's look at some examples that illustrate how to make inverse modeling work without an exact copy of the original signal.

Adaptive Equalization of a Dispersive Medium

A *dispersive* medium is one in which different frequencies travel at different velocities. That is, the *group delay* is not constant. To grasp these terms, let's say we have a medium or channel with frequency response H_ω . Response H_ω may be completely characterized

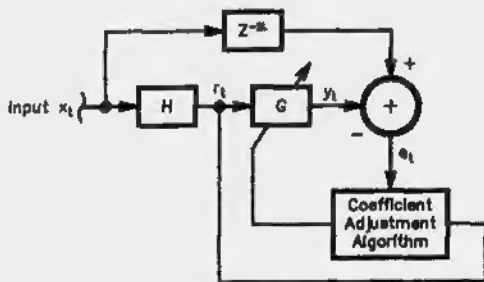


Fig 3—Block diagram of an inverse-modeling system.

by its amplitude response, A_ω and its phase response, ϕ_ω :

$$H_\omega = A_\omega e^{j\phi_\omega} \quad (\text{Eq 6})$$

The time delay (or phase delay) through the channel is:

$$t_{\text{prop}} = \frac{-\phi_\omega}{\omega} \quad (\text{Eq 7})$$

and the group delay is equal to the differential time delay:

$$t_g = \frac{-d\phi_\omega}{d\omega} \quad (\text{Eq 8})$$

A medium is said to be dispersive if the group delay is not a constant function of frequency; that is, if:

$$\frac{d^2\phi_\omega}{d\omega^2} \neq 0 \quad (\text{Eq 9})$$

Dispersive propagation is very similar to multipath, since it also implies that received information is smeared over time. Now let's say that we have a dispersive medium that is not horribly so. We also stipulate that noise levels are reasonably low, so as not to be a problem in demodulation of the data signal we're going to send through the medium. To further simplify what follows, let's also say the channel has a very large bandwidth.

Adaptive equalization may conveniently be discussed by considering the case of a single carrier, modulated by a single binary signal. While that is not a common situation on telephone lines, the format is still used over radio quite a bit. In any case, it is the simplest instance, and study of m -ary or multiple-carrier systems stems from it.

Now a simple data transmitter encodes a binary one as a transition of one polarity; a binary zero is encoded as a transition of the opposite polarity. That is true no matter the modulation format. FSK, PSK and other traditional formats may employ rapid polarity transitions that, unless otherwise limited, may cause the signal to occupy a rather large bandwidth. Even through a channel of large bandwidth, dispersion alters the shape of the transitions received because the carrier and modulation sidebands propagate at different velocities. That ultimately limits the data rate that may be supported.

Let's look at what happens when a very sharp one-zero transition passes through our dispersive channel (see Fig 5). What started out as an instantaneous state reversal now becomes smeared in time; its shape is determined by the impulse response of the channel. The group-delay-induced distortion makes recovery of the data more difficult. We can say that the

received signal is the convolution of the original signal and the impulse response of the channel. Its spectrum is the product of the spectrum of the original signal and the transfer function of the propagating medium. In other words, convolution in the time

domain is equivalent to multiplication in the frequency domain.⁵

Forward Equalization

It is often desirable to equalize the channel so that it can support higher data rates. The goal of an equalizer is

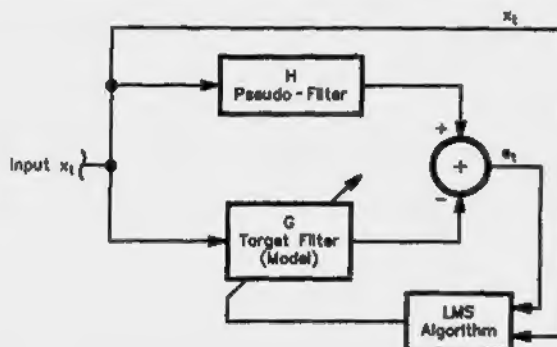


Fig 4—Block diagram of an LMS filter-design algorithm.

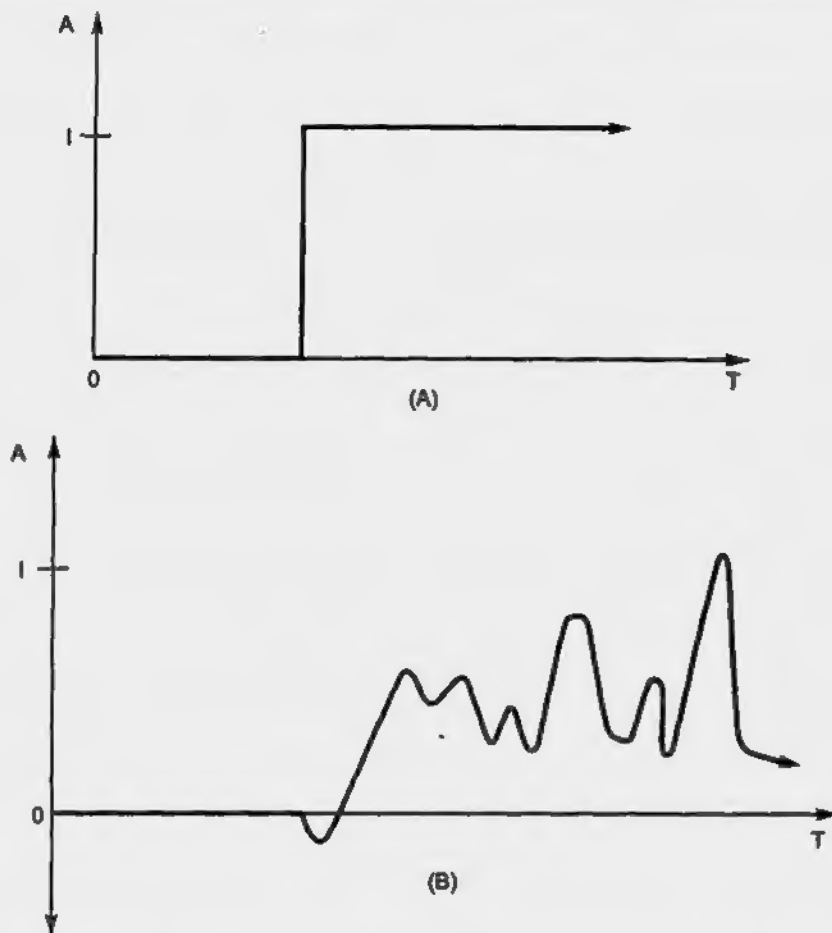


Fig 5—A: A sharp data-state transition. B: Transition as received through a dispersive medium.

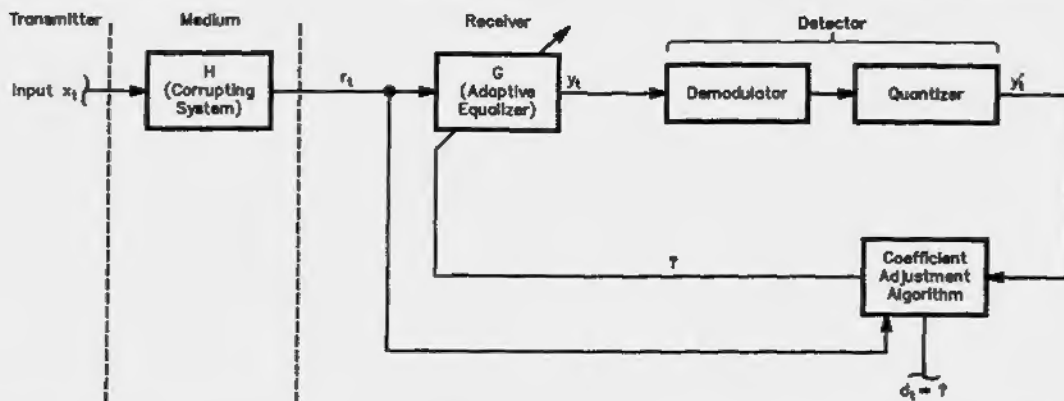


Fig 6—Block diagram of forward equalization.

to achieve a constant group delay and a flat frequency response. To equalize a channel, insert an FIR filter (G) into the data path and add a demodulator and quantizer at its output, as in Fig 6. So how do we decide how to adjust the equalizer? Well, one way is to arrange for a known *training sequence* to be transmitted and to compare the equalized signal with a locally generated copy of that sequence. The LMS algorithm may be used to adapt the equalizer. Then the block diagram is as shown in Fig 7. That system is fine for channels whose conditions do not change rapidly, as long as retraining can be tolerated periodically.

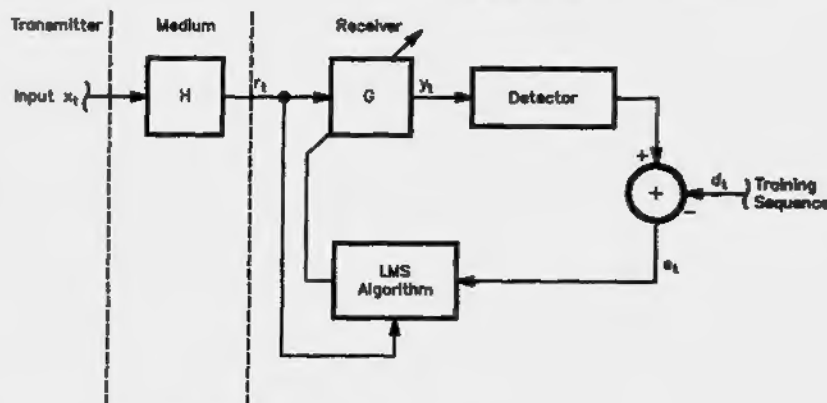


Fig 7—Forward equalization using a training sequence.

Decision Feedback Equalization

A method for deriving d_t using only the adaptive filter's output was discovered by R. W. Lucky of Bell Laboratories,⁶ obviating the need for *a priori* knowledge of the original signal. Lucky (an apt name!) found that d_t could be approximated by the demodulated signal itself, as shown in Fig 8. The bet is that if the dispersion is not severe, the demodulator generates bit decisions that are close to correct, and the adaptive filter moves toward the correct solution.

As it happens, this system works well when not much noise is present and when the dispersion is mild. Performance is improved when the sampling rate is increased beyond just once per bit.

That is fine for digital signals, but what about analog signals? The decision-making process is much tougher in that case; but the processes of *linear prediction* and *autocorrelation* may be used to steer the algorithm. Details of that are beyond the scope of

this paper. For more information, refer to Widrow and Stearns.⁷

Homomorphic Deconvolution

Now that is an esoteric phrase, but what does it mean? Well, it's a method of deconvolution that uses nonlinear transforms of signals, which are manipulated algebraically. More specifically, the nonlinear transform used is the logarithm. I'll show how a useful property of logarithms reduces multiplicative systems to simple superposition and why that is useful in deconvolving signals.

As mentioned before, convolution in the time domain is equivalent to multiplication in the frequency domain. When a signal passes through a propagation medium, the spectrum of the convolved signal is the product of the original signal's spectrum and the frequency response of the medium.

For an input signal x_t having spectrum X_ω and a medium having impulse response h_n and frequency response H_ω a convolved signal y_t has spectrum:

$$Y_\omega = X_\omega H_\omega \quad (\text{Eq 10})$$

Now for that useful property of logarithms, which is:

$$\log(ab) = \log a + \log b \quad (\text{Eq 11})$$

If we take the logarithm of Y_ω we have:

$$\begin{aligned} \log(Y_\omega) &= \log(X_\omega H_\omega) \\ &= \log(X_\omega) + \log(H_\omega) \\ &= C_{X_\omega} + C_{H_\omega} \end{aligned} \quad (\text{Eq 12})$$

Taking the inverse Fourier transform of Eq 12 therefore results in the sum of two time-domain sequences:

$$c_{y_t} = c_{x_t} + c_{h_t} \quad (\text{Eq 13})$$

c_{y_t} is called the *cepstrum* of y_t . That term and a bunch of other funny terms were coined in a paper by Bogert,

Healy and Tukey.⁸ The block diagram of a system that produces it is shown in Fig 9.

Eq 13 is a useful result since, in many reverberant environments, the two cepstral components c_{xt} and c_{ht} are easily separated because they are so different. For example, let's say that most of the energy in c_{ht} lies at low values of t and most of the energy in c_{xt} lies at high values of t . That might be the case for a voice bouncing around in the Taj Mahal. Simple window functions may segregate the individual energy contributions. Then each cepstral component may be transformed back to a regular time sequence using a process that is the inverse of Fig 9. That inverse system is shown in Fig 10 for one of the components, c_{xt} . Its output is a deconvolved (restored) version of x_t .

Homomorphic deconvolution requires minimal information about the nature of the original signal and of the propagation medium. The basic requirement is that the significant length of the medium's impulse response be considerably different from the rates of change in the original signal. Where echoes are spaced at a constant period, the contribution of c_{ht} may be removed with a window that looks like a notch filter, removing only those samples that fall within a small range of values of t . In that last case, though, a less-complex method may exist for *de-reverberation*.

A Sigma-Delta Method for De-Reverberation

In the special case where all echoes are spaced apart in time by a constant amount and those echoes decay in amplitude geometrically, a more straightforward method may be used to recover the original signal. Such reverberant environments may be found in radio communications systems and in public-address venues like large baseball or football stadia, for example. You may hear the announcer get on the public-address system and say, "Now batting, batting, batting...number nineteen, nineteen, nineteen...Tony Gwynn, Gwynn, Gwynn!"

The sound you hear is the sum of the direct signal and an infinite series of regularly spaced echoes declining in amplitude exponentially. The situation may compactly be represented as a summation:

$$y_t = \sum_{k=0}^{\infty} \mu^k (x_{t-nk}) \quad (\text{Eq 14})$$

where x_t is the input signal, μ is a positive constant less than unity and n is the number of sample times between

echoes. This is clearly a causal system since output depends only on present and past samples. The original signal may be recovered using "first-differencing" (discrete differentiation):

$$y_t - \mu y_{t-n} = \sum_{k=0}^{\infty} \mu^k (x_{t-nk}) - \mu \sum_{k=0}^{\infty} \mu^k (x_{t-nk-n}) = x_t \quad (\text{Eq 15})$$

This is approximately the difference between samples of the corrupted signal spaced one echo-time apart; but this method ignores all echoes but the first. We obviously have to wait for that first echo to occur to retrieve its energy. The system introduces a delay of n sample times before producing the desired output. Additional energy contained in all subsequent echoes is lost with this algorithm.

The total signal amplitude contributed by any particular original sample is the sum of the direct signal and all its echoes, which, assuming the original signal is of unity amplitude, is:

$$E_{\text{total}} = \sum_{k=0}^{\infty} \mu^k = 1 - \frac{\mu}{\ln \mu} \quad (\text{Eq 16})$$

That means that when $\mu=0.93$, less than one tenth of the total energy is recovered by using only the first echo—a lot of the energy is in the other echoes. Signal-to-noise ratio (S/N) would be degraded by about 20 dB. In this case, it's clearly worth an additional wait to improve our lot.

To recoup the energy in all echoes would take an infinitely long period, so

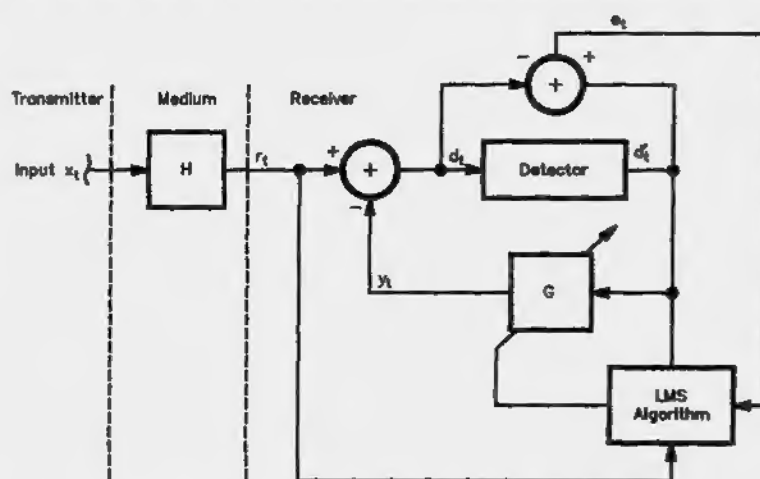


Fig 8—Decision feedback equalization using the received data as a desired signal. In this method, an adaptively filtered copy of the detected signal is subtracted from the unmodified received signal to cancel intersymbol interference. D' is the output. Feedback equalization is typically used in concert with feed-forward equalization. For more detail, see Reference 7.

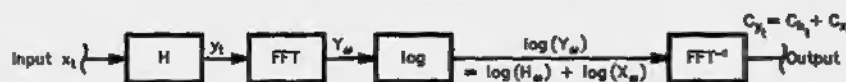


Fig 9—Block diagram of a cepstral transform.



Fig 10—Block diagram of an inverse cepstral transform.

we can only regain the energy over some number of echoes, L , for which we are willing to wait. The delay incurred is nL sampling periods. To recover the energy contained in echoes beyond the first, consider taking "second-differencing," or the weighted difference between samples two echo times apart. We have:

$$\begin{aligned} y_t - \mu^2 y_{t-2n} &= \sum_{k=0}^{L-1} \mu^k (x_{t-nk}) - \mu^2 \sum_{k=0}^{L-1} \mu^k (x_{t-nk-2n}) \\ &= x_t + \mu x_{t-n} \end{aligned} \quad (\text{Eq 17})$$

$$\therefore \mu x_{t-n} = y_t - \mu^2 y_{t-2n} - x_t$$

where x_t is determined by Eq 15 above. x_{t-n} was x_t n samples ago, and is added to the result of Eq 17 to yield $x_{t-n} + \mu x_{t-n}$. A similar operation may be performed for $y_t - \mu^3 y_{t-3n}$, $y_t - \mu^4 y_{t-4n}$, and so on, continuously, to build energy from echoes as they get older. In that way, almost all the energy can be regained from a reverberant environment having a single, uniform echo.

Performance of the Sigma-Delta Method

Over a finite number of echo intervals, L (during which we wait nL sample times), the energy recovered is not as much as in the infinite summations. It is only:

$$\begin{aligned} E_L &= \sum_{k=0}^{L-1} \mu^k \\ &= \left(1 - \frac{\mu}{\ln \mu}\right) + \frac{\mu^L}{L \ln \mu} \end{aligned} \quad (\text{Eq 18})$$

If $\mu=0.93$ and $L=8$, the S/N degradation would be about 1 dB, since about 88% of the energy would have been recovered.

The algorithm counts on absolute frequency and phase accuracy between transmitter and receiver. Serious phase distortion or frequency errors would render the sigma-delta method unusable. It is not well suited to SSB operation, therefore, without a pilot carrier and a synchronous (phase-locked) receiver, or other suitable demodulators.

The algorithm is also quite sensitive to phase noise in the local oscillators of radio transceivers and to dispersive

propagation—the thing so similar to the problem it attempts to solve! So this algorithm turns out not to be a very good pick at all, but it is relatively simple compared to homomorphic processing.

Summary

This article showed convenient modeling methods for reverberant and dispersive environments. Systems for deconvolution were discussed that correct for multipath and dispersion; they even produce a model of the corrupting system in most cases. In some instances, the model of the corrupting system may be the thing that is sought. That is the case in ionospheric studies or in planetary science, where the impulse response of the model represents a map of the atmosphere or planetary surface, respectively. Deconvolution systems are sometimes adaptive and thus are capable of handling changing propagation conditions. Homomorphic deconvolution is generally not adaptive and relies on some knowledge of the differences between the desired signal and the nature of the medium.

Research is ongoing to use adaptive receiving arrays and homomorphic processing on weak, convolved signals. Moonbounce (EME) modes are a particular target of that research.

Notes

1. C. Hutchinson, Editor, *The 2001 ARRL Handbook*, p 18.5.
2. A. Oppenheim and R. Schaffer, *Digital Signal Processing*, (Englewood Cliffs, New Jersey: Prentice-Hall, 1975).
3. B. Widrow and M. Hoff, Jr., "Adaptive switching circuits," *IRE WESCON Convention Records*, Part 4, IRE, 1960.
4. S. Reyer and D. Herschberger, "Using the LMS Algorithm for QRM and QRN Reduction," *QEX*, Sep 1992, pp 3-8.
5. M. Frerking, *Digital Signal Processing in Communications Systems*, (New York: Van Nostrand-Reinhold, 1994) p 11.
6. R. Lucky, "Techniques for Adaptive Equalization of Digital Communications Systems," *Bell Systems Technical Journal*, volume 45, pp 255-286, Feb, 1966.
7. B. Widrow and S. Stearns, *Adaptive Signal Processing*, (Prentice-Hall, 1985).
8. B. Bogert, M. Healy and J. Tukey, "The Frequency Analysis of Time Series for Echoes: Cepstrum, Pseudo-autocovariance, Cross-Cepstrum and Saphe Cracking," *Proceedings of the Symposium on Time Series Analysis*, (New York: John Wiley and Sons, 1963) pp 209-243. □